

# **Dokumentacija REST API v 3.0**

**Kraków, 1 sierpnia 2016**

**FreshMail Sp. z o.o.**, Al. 29 Listopada 155 c, 31-406 Kraków  
tel. +48 12 617 61 40, [info@freshmail.pl](mailto:info@freshmail.pl), [freshmail.pl](http://freshmail.pl)

## Spis treści

<b>Opis API.....</b>	<b>4</b>
Uwierzytelnienie.....	4
Odpowiedzi API.....	6
Obsługa błędów.....	6
Ścieżka dostępu.....	7
Ogólne kody błędów.....	7
<b>Opis metod używanych w API.....</b>	<b>8</b>
Ping – testowanie.....	8
Maile transakcyjne – wysyłanie pojedynczych maili.....	9
Wiadomości SMS.....	11
Uwagi ogólne.....	11
Rodzaje wiadomości SMS.....	12
Wysyłanie wiadomości SMS.....	13
Raporty – raporty z kampanii.....	14
Pobieranie listy wysłanych kampanii.....	14
Pobieranie zbiorczego raportu z kampanii.....	15
Pobieranie zachowań subskrybentów w czasie.....	16
Kampanie – obsługa kampanii.....	18
Tworzenie nowej kampanii.....	18
Wysłanie kampanii.....	20
Kasowanie kampanii.....	22
Wysłanie kampanii testowej.....	23
Wysłanie kampanii.....	24
Subskrybenci – zarządzanie subskrybentami.....	25
Kody statusów subskrybentów.....	25
Dodanie subskrybenta .....	26
Modyfikowanie danych subskrybenta.....	27
Pobieranie danych subskrybenta.....	28
Kasowanie subskrybenta.....	29
Subskrybenci – zarządzanie wieloma subskrybentami w pojedynczym żądaniu.....	30
Dodawanie wielu subskrybentów.....	30
Modyfikowanie danych wielu subskrybentów.....	32
Modyfikowanie pól dodatkowych wszystkich subskrybentów.....	33
Pobieranie danych wielu subskrybentów.....	35
Kasowanie wielu subskrybentów.....	36

Tworzenie kont.....	37
Listy subskrypcyjne - zarządzanie.....	39
Tworzenie list.....	39
Modyfikowanie list.....	40
Usuwanie list.....	41
Pobieranie danych o listach.....	42
Tworzenie pól dodatkowych w listach.....	43
Wyświetlenie listy pól dodatkowych w podanej liście.....	44
Spamtesty.....	45
Sprawdzenie zgodności z filtrami SpamAssassin.....	45

**Wersja dokumentu:** 1.0.22

**Autorzy:** Tadeusz Kania,  
Piotr Suszalski,  
Grzegorz Gorczyca

**Kontakt:** [api@freshmail.pl](mailto:api@freshmail.pl)

## Opis API

API systemu FreshMail zostało zrealizowane według wzorca REST. Komunikacja polega na wysłaniu żądania HTTP pod adres <https://api.freshmail.com/rest/> i ewentualnym przestaniu wymaganych danych. Odpowiedź jest zwracana w formacie JSON.

## Uwierzytelnienie

Każde wysłane żądanie po API wymaga uwierzytelnienia. Uwierzytelnienie następuje poprzez wysłanie nagłówków HTTP o nazwach:

- X-Rest-ApiKey** zawiera unikalny klucz dostępu do API, każdy z klientów posiada swój unikalny klucz API.  
subskrypcje → lista subskrypcyjna → opcje zaawansowane i API
- X-Rest-ApiSign** zawiera podpis żądania. Podpis jest generowany na podstawie parametrów wysyłanych w żądaniu oraz na podstawie ApiSecret.

Istnieją dwa schematy generowania **X-Rest-ApiSign**, oba zostały przedstawione poniżej:

Schemat oparty na **URL-encoded query**:

wymagane jest dodanie nagłówka Content-Type w żądaniu http o wartości:

text/html lub application/xml lub application/json, np.:

Content-Type: application/json

wtedy można wygenerować **X-Rest-ApiSign** w oparciu o algorytm:

$\text{sha1}(\text{ApiKey} + \text{GET\_data} + \text{POST\_data} + \text{ApiSecret})$

gdzie:

- ApiKey** 32 znakowy Api key.
- GET\_data** pełna ścieżka dostępu – np. /rest/ping, /rest/mail itp.  
Więcej szczegółów w sekcji „*Ścieżka dostępu*”.
- POST\_data** dane POST umieszczone w żądaniu. Jeśli w żądaniu znajdują się tylko dane GET to wartość POST\_data jest pusta. Wartości muszą być escapowane na potrzeby URL, np.: email=test%40test.pl&subject=test+emaila
- ApiSecret** 40 znaków, unikalny sekret dla każdego z klientów. **W przypadku skompromitowania klucza należy bezzwłocznie wygenerować nowy.**

Schemat oparty na **JSON**:

wymagane jest dodanie nagłówka Content-Type w żądaniu http w postaci:

Content-Type: application/json

wtedy można wygenerować **X-Rest-ApiSign** w oparciu o algorytm:

sha1( ApiKey + GET\_data + JSON\_data + ApiSecret )

gdzie:

<b>ApiKey</b>	32 znakowy Api key.
<b>GET_data</b>	pełna ścieżka dostępu – np. /rest/ping, /rest/mail itp. Więcej szczegółów w sekcji „ <i>Ścieżka dostępu</i> ”.
<b>JSON_data</b>	dane w formacie JSON jakie są wysyłane w żądaniu. Jeśli w żądaniu znajdują się tylko dane GET to wartość JSON_data jest pusta. Przykładowa wartość JSON_data: {"subscriber":"test@test.pl"}
<b>ApiSecret</b>	40 znaków, unikalny sekret dla każdego z klientów. <b>W przypadku skompromitowania klucza należy bezzwłocznie wygenerować nowy.</b>

## **Odpowiedzi API**

Odpowiedź na każde żądanie wysłane do API jest w formacie JSON, np.

```
{"status":"OK","data":[1,2,3]}
```

Jeśli zapytanie powiedzie się wtedy pod kluczem „**status**” będziemy mieli wartość 'OK'. Jeśli metoda ma zwrócić dane będą one pod kluczem „**data**”. Natomiast jeśli wystąpi błąd pod kluczem „**status**” będzie wartość „**ERROR**”, a pod kluczem „**errors**” znajdzie się opis błędów jakie wystąpiły wraz z kodami błędów. W przypadku błędu kod HTTP zwracany w nagłówku jest adekwatny do zaistniałego błędu, czyli np. **404**, **403** itp. W przypadku braku błędu kod HTTP to **200**.

## **Obsługa błędów**

W razie wystąpienia błędu status odpowiedzi jest równy „**ERROR**” natomiast pole „**errors**” zawiera opis słowny błędu oraz kod błędu, np.

```
{"errors":[
  {"message":"Brak tematu email",
   "code": 1202}
],
"status":"ERROR"}
```

lub

```
{"errors":[
  {"message": "Błędny adres odbiorcy",
   "code": 1201},
  {"message": "Brak tematu email",
   "code": 1202}
],
"status":"ERROR"}
```

## **Ścieżka dostępu**

Każda z akcji jaką można wykonać ma swoją osobną ścieżkę dostępu. Ścieżka zbudowana jest w następujący sposób:

`/rest/[Controller]/[Action]/[param]/[param]/[param]`

<b>Controller</b>	nazwa kontrolera np. ping, mail, itp.
<b>Action</b>	nazwa akcji w obrębie kontrolera.
<b>param</b>	parametr, nie może zawierać znaku „/”, w większości przypadków dodatkowe parametry GET są zbędne. Jeśli są wymagane to ilość oraz ich format jest podany w opisie akcji.

## **Ogólne kody błędów**

<b>500</b>	Błąd serwera (Internal Server Error).
<b>1000</b>	Brak autoryzacji.
<b>1001</b>	Nie znaleziono kontrolera / akcji.
<b>1002</b>	Nieobsługiwany protokół. Proszę skorzystać z protokołu https.
<b>1003</b>	Nieobsługiwany rodzaj żądania, np. wysłano GET, a oczekiwano POST.
<b>1004</b>	Brak uprawnień do wywołania danego kontrolera/akcji



## ***Opis metod używanych w API***

### ***Ping – testowanie***

Kontroler ten służy do przetestowania poprawności działania API. Osobno można przetestować poprawność działania żądań typu GET oraz żądań typu POST.

Wywołanie: **GET /rest/ping**

W odpowiedzi serwer zwróci „pong”.

Wywołanie: **POST /rest/ping**

W odpowiedzi serwer zwróci dane które wysła się w POST.

## Maile transakcyjne – wysyłanie pojedynczych maili

Kontroler służy do wysyłki pojedynczych maili transakcyjnych. Domyślnie wszystkie parametry powinny być przekazane w kodowaniu **UTF-8**. Można to zmienić korzystając z parametru **encoding**.

Wywołanie: **POST /rest/mail**

Dane w POST:

<b>subscriber</b>	Wymagane	Adres email subskrybenta do którego wysyłamy email.
<b>subject</b>	Wymagane	Tytuł emaila.
<b>html</b>	Wymagane*	Treść html emaila (wymagana jeśli nie ma treści text).
<b>text</b>	Wymagane*	Treść tekstowa emaila (wymagana jeśli nie ma treści html).
<b>from</b>	Opcjonalne	Pole „Od”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
<b>from_name</b>	Opcjonalne	Pole „Nazwa nadawcy”, jeśli będzie puste w to miejsce wstawi się domyślna nazwa nadawcy z systemu.
<b>reply_to</b>	Opcjonalne	Pole „Odpowiedz do”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
<b>encoding</b>	Opcjonalne	Kodowanie w jakim ma zostać wysłany email. Domyślnie jest to UTF-8. Pamiętaj, że email musi być przekazany w tym samym kodowaniu, które jest przekazane w parametrze encoding.  Obsługiwane kodowania to: UTF-8, iso-8859-2
<b>attachments</b>	Opcjonalne	Link do załącznika, który ma zostać dołączony do maila. Maksymalny rozmiar załączników to 0.5 MB
<b>tracking</b>	Opcjonalne	Pole przyjmujące wartość 0/1. Jeśli przekazane zostanie 1 to email będzie trackingowany przez kody śledzące zachowania subskrybenta.  <b>Uwaga: oznacza to, że domena w jakiej pojawiają się linki zostanie zmieniona oraz do wersji HTML zostanie dodany obrazek śledzący.</b>
<b>domain</b>	Opcjonalnie	Domena do rebrandingu, wartość aktywna tylko jeśli jest włączony dla maila tracking. Jeśli wartość będzie pusta lub nie zostanie przekazana zostanie umieszczona domyślna domena do rebrandingu. Przykładowa domena: mojanazwafirmy.mailnews.pl  <b>Uwaga: aby korzystać z domen do rebrandingu w pierwszej kolejności należy taką domenę zgłosić korzystając z panelu w aplikacji.</b>
<b>tag</b>	Opcjonalnie	Oznaczenie typu wysłanego maila. Wykorzystując tagi, raporty z maili transakcyjnych zostaną podzielone na podgrupy.  Przykładowe tagi: Mail Akceptacyjny, Mail przypominający, itp.

## Kody błędów:

- 1201** Błędny adres subskrybenta.
- 1202** Brak tematu maila.
- 1203** Brak treści HTML oraz tekstowej.
- 1204** Błędny adres „odpowiedz do”.
- 1205** Błędny adres email nadawcy.
- 1206** Nazwa nadawcy nie może być pusta.
- 1207** Przekroczono dzienny limit wysłanych maili.
- 1208** Nieobsługiwane kodowanie.
- 1209** Nieprawidłowy link do załącznika.
- 1210** Plik załącznika nie istnieje.
- 1211** Przekroczono maksymalny rozmiar załączników.
- 1212** Niepoprawna domena do rebrandingu, sprawdź czy domena została aktywowana.
- 1213** Tag kampanii niepoprawny (za długi lub zawierający niedozwolone znaki)
- 1250** Nie udało się wysłać emaila, błąd systemu.
- 1251** Nie wysłano emaila do tego subskrybenta, zablokowany administracyjnie

## Wiadomości SMS

Kontroler służy do wysyłki pojedynczych wiadomości SMS. Domyślnie wszystkie parametry muszą być przekazane w kodowaniu **UTF-8**.

### Uwagi ogólne

Numery telefonu muszą być podawane w **formacie 9 lub 11 cyfrowym** (np. 48502602702 lub 502602702), możliwe jest także dodanie znaku „+” przed prefixem kraju (np. +48502602702).

Treść pojedynczej wiadomości to standardowo do 160 znaków lub 70 znaków w przypadku wystąpienia chociaż jednego znaku specjalnego. Maksymalna długość wiadomości wynosi 459 znaków (lub 200 ze znakami specjalnymi) i jest wysłana jako 3 połączone SMS-y.

**Znakami specjalnymi** są wszystkie znaki nie mieszczące się w zestawie znaków:

@€\$¥èéùìòçøå\_^{ } \ [ ~ ] | Æ æ ß É ! " # ¨ % & ' ( ) \* + , - . / 0 - 9 ; : < = > ? A - Z Ä Ö Ñ Ü Š ž a - z ä ö ñ ü à <enter>

### Uwaga!

Znaki: ^ { } [ ] ~ \ | <enter> zawsze liczone są podwójnie (bez względu na to czy wysyłamy SMS ze znakami specjalnymi czy bez) ze względu na wymogi specyfikacji GSM.

SMS bez znaków specjalnych		SMS ze znakami specjalnymi	
Maksymalna ilość znaków	Ilość SMSów	Maksymalna ilość znaków	Ilość SMSów
160	1	70	1
306	2	134	2
459	3	200	3

## Rodzaje wiadomości SMS

Bramka umożliwia wysłanie wiadomości dwóch rodzajów: wiadomości typu **ECO** oraz wiadomości typu **PRO**.

**SMS typu ECO** to wiadomość wysyłana z losowego 9-cio cyfrowego numeru telefonu.

**SMS typu PRO** to wiadomość wysyłana z nadpisem alfanumerycznym, czyli zdefiniowanym polem nadawcy. Pole to nie może zawierać więcej niż 11 znaków z zestawu a-zA-Z0-9 oraz znak myślnika „-”. Aby korzystać z SMS typu PRO konieczne jest **zgłoszenie nadpisu**.

Aby móc korzystać z bramki SMS należy w pierwszej kolejności podpisać umowę. Ceny za sms typu **ECO** oraz typu **PRO** są zawarte w umowie.

## Wysyłanie wiadomości SMS

Wywołanie: **POST /rest/sms**

Dane w POST:

<b>gsm</b>	Wymagane	Numer telefonu
<b>text</b>	Wymagane	Treść SMSa. Więcej szczegółów w sekcji „ <b>Uwagi ogólne</b> ”
<b>from</b>	Opcjonalne	Nadpis / nadawca. Więcej szczegółów w sekcji „ <b>Rodzaje wiadomości SMS</b> ”. Jeśli nadpis będzie poprawny zostanie wysłany sms typu <b>PRO</b>
<b>single</b>	Opcjonalnie	Ograniczenie długości SMSa. Jest to informacja czy do subskrybenta ma być wysłany SMS składający się z więcej niż jednej części. Możliwe wartości to <b>0</b> oraz <b>1</b> . Jeśli to pole będzie puste domyślnie przyjmuje wartość 0. Jeśli przekazana zostanie wartość 1 to zostanie zwrócony błąd w momencie w którym wiadomość będzie za długa aby wystać ją w jednej wiadomości SMS.

Kody błędów:

<b>1101</b>	Numer GSM jest błędny
<b>1102</b>	Treść SMSa jest pusta
<b>1103</b>	Treść SMSa jest za długa (SMS bez znaków specjalnych)
<b>1104</b>	Treść SMSa jest za długa (SMS ze znakami specjalnymi)
<b>1105</b>	Nadawca jest błędny
<b>1106</b>	Nadawca został niepotwierdzony (nadawca jest poprawny, ale oczekuje na weryfikację)
<b>1107</b>	Żądanie jest w złym kodowaniu, obsługiwane kodowanie to UTF-8
<b>1108</b>	Treść SMSa jest za długa (SMS bez znaków specjalnych z opcją single=1)
<b>1109</b>	Treść SMSa jest za długa (SMS ze znakami specjalnymi z opcją single=1)
<b>1150</b>	SMS nie zostanie wysłany, konieczne jest podpisanie umowy

## ***Raporty – raporty z kampanii***

Kontroler służy do pobierania raportów oraz list wysłanych kampanii.

### **Pobieranie listy wysłanych kampanii**

Wywołanie: **GET /rest/reports/campaignsList/[page]**

Dane w GET:

<b>page</b>	Opcjonalne	Numer strony z kampaniami, domyślnie pobierana jest pierwsza strona
-------------	------------	---

W odpowiedzi serwer zwróci listę 25 kampanii. Zwracane dane:

<b>name</b>	Nazwa kampanii
<b>topic</b>	Temat emaila w kampanii
<b>subscribers</b>	Ilość subskrybentów do których była wysyłka
<b>id_hash</b>	Unikalny identyfikator kampanii
<b>sent</b>	Data wysyłki kampanii, format: rrrr-mm-dd hh:ii:ss

## Pobieranie zbiorczego raportu z kampanii

Wywołanie: **GET /rest/reports/campaign/[id\_hash]**

Dane w GET:

<b>id_hash</b>	Wymagane	Unikalny identyfikator kampanii, można go otrzymać wywołując metodę /rest/reports/campaignsList
----------------	----------	---

Zwracane dane:

<b>subscribers</b>	Liczba wszystkich subskrybentów do których wysłano kampanię
<b>delivered</b>	Liczba dostarczonych wiadomości
<b>hard_bounce</b>	Liczba wiadomości odbitych „twardo” ( <a href="http://www.ietf.org/rfc/rfc1893.txt">http://www.ietf.org/rfc/rfc1893.txt</a> )
<b>soft_bounce</b>	Liczba wiadomości odbitych „miętko” ( <a href="http://www.ietf.org/rfc/rfc1893.txt">http://www.ietf.org/rfc/rfc1893.txt</a> )
<b>opened</b>	Liczba wszystkich otwartych wiadomości
<b>clicked</b>	Liczba wszystkich klikniętych linków
<b>unique_opened</b>	Liczba unikalnych otwarć wiadomości
<b>unique_clicked</b>	Liczba unikalnych klikniętych wiadomości
<b>resigned</b>	Liczba wypisanych subskrybentów



## Pobieranie zachowań subskrybentów w czasie

Wywołanie: **GET /rest/reports/campaignTimeDetails/[id\_hash]**

Dane w GET:

<b>id_hash</b>	Wymagane	Unikalny identyfikator kampanii, można go otrzymać wywołując metodę /rest/reports/campaignsList
----------------	----------	---

Zwracane dane – tablica z kolejnymi działaniami subskrybentów w czasie:

```
{"status":"OK","data":  
  [  
    {"opened":"4",  
     "unique_opened":"4",  
     "clicked":"3",  
     "unique_clicked":"1",  
     "time":"2012-03-26 16:50"},  
    {"opened":"1",  
     "unique_opened":"1",  
     "clicked":"1",  
     "unique_clicked":"1",  
     "time":"2012-03-27 08:30"}  
  ]  
}
```

<b>opened</b>	Liczba wszystkich otwarć w danym przedziale czasowym
<b>clicked</b>	Liczba wszystkich kliknięć w danym przedziale czasowym
<b>unique_opened</b>	Liczba unikalnych otwartych emaili w danym przedziale czasowym
<b>unique_clicked</b>	Liczba unikalnych emaili z klikniętym linkiem w danym przedziale czasowym
<b>time</b>	Czas w formacie „rrrr-mm-dd HH:ii”, rozdzielczość okna to 10 minut

## Kody błędów:

- 1401** Niepoprawny numer strony
- 1402** Kampania o podanych id\_hash nie istnieje

## ***Kampanie – obsługa kampanii***

Kontroler służy do tworzenia, sprawdzania i wysyłania kampanii.

### **Tworzenie nowej kampanii**

Wywołanie: **POST /rest/campaigns/create**

Dane w POST:

<b>name</b>	Wymagane	Nazwa kampanii
<b>url</b>	Opcjonalne	Adres strony z której ma zostać pobrany kod HTML
<b>html</b>	Wymagane*	Treść html emaila (wymagana jeśli nie ma treści text oraz adresu url).
<b>text</b>	Wymagane*	Treść tekstowa emaila (wymagana jeśli nie ma treści html).
<b>subject</b>	Opcjonalne	Tytuł jaki ma mieć email wysłany przez system. W razie braku jako tytuł będzie wzięta nazwa kampanii.
<b>from_address</b>	Opcjonalne	Pole „Od”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
<b>from_name</b>	Opcjonalne	Pole „Nazwa nadawcy”, jeśli będzie puste w to miejsce wstawi się domyślna nazwa nadawcy z systemu.
<b>reply_to</b>	Opcjonalne	Pole „Odpowiedz do”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
<b>list</b>	Wymagane*	Hash listy subskrypcyjnej do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jednej listy. W przypadku wysyłki do grupy nie trzeba podawać listy.
<b>group</b>	Wymagane*	Hash grupy do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jednej grupy. W przypadku wysyłki do listy subskrypcyjnej nie trzeba podawać grupy.
<b>resignlink</b>	Opcjonalne	Adres URL na jaki zostanie przekierowany odbiorca po kliknięciu na link rezygnacji

Zwracane dane:

**hash** W przypadku poprawnego utworzenia kampanii zwrócony zostaje jej hash

## Kody błędów:

- 1701** Brak nazwy kampanii
- 1702** Brak treści (nie podano treści text, html ani url )
- 1703** Błędny adres url
- 1704** Brak klienta (błąd wewnętrzny)
- 1705** Brak parametrów (błąd wewnętrzny)
- 1706** Błędny adres nadawcy (from\_address)
- 1707** Błędny adres „reply to”
- 1708** Brak listy subskrypcyjnej i grupy
- 1709** Co najmniej jeden hash listy jest błędny.
- 1710** Co najmniej jeden hash grupy jest błędny.
- 1711** Nie znaleziono list o podanym hashu.
- 1712** Nie znaleziono grupy o podanym hashu.
- 1713** Link rezygnacji musi być poprawnym adresem URL

## Wysłanie kampanii

Wywołanie: **POST /rest/campaigns/edit**

Dane w POST:

<b>id_hash</b>	Wymagane	Hash kampanii do edytowania
<b>name</b>	Opcjonalne	Nazwa kampanii
<b>url</b>	Opcjonalne	Adres strony z której ma zostać pobrany kod HTML
<b>html</b>	Opcjonalne	Treść html emaila
<b>text</b>	Opcjonalne	Treść tekstowa emaila
<b>subject</b>	Opcjonalne	Tytuł jaki ma mieć email wysłany przez system. W razie braku jako tytuł będzie wzięta nazwa kampanii.
<b>from_address</b>	Opcjonalne	Pole „Od”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
<b>from_name</b>	Opcjonalne	Pole „Nazwa nadawcy”, jeśli będzie puste w to miejsce wstawi się domyślna nazwa nadawcy z systemu.
<b>reply_to</b>	Opcjonalne	Pole „Odpowiedz do”, jeśli będzie puste w to miejsce wstawi się domyślny adres nadawcy z systemu.
<b>list</b>	Opcjonalne	Hash listy subskrypcyjnej do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jednej listy. W przypadku wysyłki do grupy nie trzeba podawać listy.
<b>Group</b>	Opcjonalne	Hash grupy do której ma być wysłana kampania. Może być to tablica hashy jeśli wysyłka będzie do więcej niż jednej grupy. W przypadku wysyłki do listy subskrypcyjnej nie trzeba podawać grupy.
<b>resignlink</b>	Opcjonalne	Adres URL na jaki zostanie przekierowany odbiorca po kliknięciu na link rezygnacji

### Uwaga:

W żądaniu można podać tylko te parametry które chcemy aby uległy zmianie, jeśli dodamy parametr **list** lub **group** (nie można podać ich obu naraz) to stare ustawienie grup/list subskrypcyjnych zostanie całkowicie nadpisane.

Kody błędów:

- 1750** Brak kampanii do edycji o podanym hashu
- 1751** Kampania już została ustawiona do wysyłki, modyfikacja niemożliwa

Reszta kodów błędów jest identyczna jak w przypadku tworzenia kampanii.

## Kasowanie kampanii

Wywołanie: **POST /rest/campaigns/delete**

Dane w POST:

<b>hash</b>	Wymagane	Hash kampanii do skasowania
-------------	----------	-----------------------------

Kody błędów:

<b>1724</b>	Brak kampanii o takim hashu
<b>1798</b>	Kampania już została skasowana

## Wysłanie kampanii testowej

Wywołanie: **POST /rest/campaigns/sendTest**

Dane w POST:

<b>hash</b>	Wymagane	Hash kampanii którą chcemy wysłać
<b>emails</b>	Wymagane	Adres lub tablica adresów na które test ma być wysłany.
<b>custom_fields</b>	Opcjonalne	Tablica z polami dodatkowymi, w przypadku pozostawienia pola pustego wartości pól dodatkowych zostaną uzupełnione losowymi danymi z listy subskrypcyjnej

Przykładowe żądanie:

```
{"hash":"hash_kampanii",  
"emails":"test@test.pl",  
"custom_fields":{  
  "imie":"Jan",  
  "nazwisko":"Kowalski"}}
```

lub

```
{"hash":"hash_kampanii",  
"emails":"test1@test.pl","test2@test.pl"}}
```

Kody błędów:

<b>1721</b>	Brak hasha lub jest on niepoprawny.
<b>1722</b>	Brak adresu email.
<b>1723</b>	Co najmniej jeden adres email jest niepoprawny.
<b>1724</b>	Kampania o takim hashu nie istnieje.
<b>1725</b>	Kampania nie jest jeszcze gotowa do wysyłki testowej.
<b>1726</b>	Wysyłka nie powiodła się.

## Wysłanie kampanii

Wywołanie: **POST /rest/campaigns/send**

Dane w POST:

<b>hash</b>	Wymagane	Hash kampanii którą chcemy wysłać
<b>time</b>	Opcjonalne	Data wysłania kampanii w formacie YYYY-MM-DD H:i:s

Kody błędów:

- 1731** Brak hasha lub jest on niepoprawny.
- 1732** Czas wysyłki jest niepoprawny
- 1733** Czas wysyłki nie może być wcześniejszy niż obecna godzina.
- 1734** Kampania o takim hashu nie istnieje.
- 1735** Kampania nie jest jeszcze gotowa do wysyłki testowej.
- 1736** Kampania już jest ustawiona do wysyłki lub jest w jej trakcie.



## ***Subskrybenci – zarządzanie subskrybentami***

Kontroler służy do dodawania i aktualizowania danych o subskrybentach w listach subskrypcyjnych.

### **Kody statusów subskrybentów**

<b>Opis słowny</b>	<b>Wartość</b>	<b>Objaśnienie</b>
<b>Aktywny</b>	<b>1</b>	Subskrybent aktywny do którego mogą być wysyłane kampanie
<b>Do aktywacji</b>	<b>2</b>	Subskrybent który musi kliknąć w link akceptacyjny w mailu potwierdzającym jego dodanie do listy
<b>Nieaktywowany</b>	<b>3</b>	Subskrybent który po okresie kilku dni nie kliknął w link akceptacyjny w mailu potwierdzającym
<b>Wypisany</b>	<b>4</b>	Subskrybent który wypisał się z subskrypcji
<b>Odbijający „miętko”</b>	<b>5</b>	Subskrybent którego skrzynka kilka razy z rzędu zwróciła maila z odpowiednim kodem ( <a href="http://www.ietf.org/rfc/rfc1893.txt">http://www.ietf.org/rfc/rfc1893.txt</a> )
<b>Odbijający „twardo”</b>	<b>8</b>	Subskrybent którego skrzynka kilka razy z rzędu zwróciła maila z odpowiednim kodem ( <a href="http://www.ietf.org/rfc/rfc1893.txt">http://www.ietf.org/rfc/rfc1893.txt</a> )

## Dodanie subskrybenta

Metoda dodaje subskrybenta jeśli nie było go w liście subskrypcyjnej lub aktualizuje jego status i dane jeśli nie był w statusie „**Aktywny**” oraz „**Do aktywacji**”

Wywołanie: **POST /rest/subscriber/add**

Dane w POST:

<b>email</b>	Wymagane	Adres email subskrybenta
<b>list</b>	Wymagane	Hash list subskrypcyjnej
<b>state</b>	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste zostanie nadany status „ <b>Do aktywacji</b> ”, lista dostępnych statusów w dziale „ <b>Kody Statusów Subskrybentów</b> ”
<b>confirm</b>	Opcjonalne	Czy do subskrybenta ma być wysłany email potwierdzający. Możliwe wartości to <b>0</b> oraz <b>1</b> . Jeśli pole będzie puste zostanie wysłany email z potwierdzeniem. To pole jest ignorowane (email nie zostanie wysłany) jeśli nadamy subskrybentowi status inny niż „ <b>Do aktywacji</b> ”.
<b>custom_fields</b>	Opcjonalne	Tablica z polami dodatkowymi i wartościami w postaci: { "tag personalizacji pola 1": "wartość", "tag personalizacji pola 2": "wartość" }

Kody błędów:

- 1301** Adres email jest niepoprawny
- 1302** Lista subskrypcyjna nie istnieje lub brak hash'a listy
- 1303** Jedno lub więcej pól dodatkowych jest niepoprawne
- 1304** Subskrybent już istnieje w tej liście subskrypcyjnej i ma status **Aktywny** lub **Do aktywacji**
- 1305** Próbowano nadać niepoprawny status subskrybenta

## Modyfikowanie danych subskrybenta

Metoda aktualizuje status i dane subskrybenta

Wywołanie: **POST /rest/subscriber/edit**

Dane w POST:

<b>email</b>	Wymagane	Adres email subskrybenta
<b>list</b>	Wymagane	Hash list subskrypcyjnej
<b>state</b>	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste aktualny status subskrybenta nie zostanie zmieniony
<b>custom_fields</b>	Opcjonalne	Tablica z polami dodatkowymi i wartościami w formacie: { "tag personalizacji pola 1": "wartość", "tag personalizacji pola 2": "wartość" }

Kody błędów:

<b>1331</b>	Subskrybent nie istnieje
<b>1302</b>	Lista subskrypcyjna nie istnieje lub brak hash'a listy
<b>1305</b>	Próbowano nadać niepoprawny status subskrybenta

## Pobieranie danych subskrybenta

Wywołanie: **GET /rest/subscriber/get/{list}/{email}**

Dane w GET:

<b>email</b>	Wymagane	Adres email subskrybenta
<b>list</b>	Wymagane	Hash list subskrypcyjnej

Zwracane dane:

<b>email</b>	Adres email subskrybenta
<b>state</b>	Status subskrybenta (zgodny z opisem w dziale „ <b>Kody Statusów Subskrybentów</b> ”)
<b>custom_fields</b>	Ilość subskrybentów do których była wysyłka
<b>sent</b>	Data wysyłki kampanii, format: rrrr-mm-dd hh:ii:ss

Kody błędów:

<b>1311</b>	Adres email jest niepoprawny
<b>1312</b>	Lista subskrypcyjna nie istnieje

## Kasowanie subskrybenta

Metoda kasuje subskrybenta z listy subskrypcyjnej

Wywołanie: **POST /rest/subscriber/delete**

Dane w POST:

<b>email</b>	Wymagane	Adres email subskrybenta
<b>list</b>	Wymagane	Hash list subskrypcyjnej

Kody błędów:

- 1321** Adres email nie istnieje w tej liście subskrypcyjnej
- 1322** Lista subskrypcyjna nie istnieje lub brak hash'a listy

## Pobieranie historii akcji subskrybenta

Metoda pobiera ostatnie akcje subskrybenta.

Wywołanie: POST **/rest/subscriber/getHistory**

Dane w POST:

<b>email</b>	Wymagane	Adres email subskrybenta
<b>list</b>	Wymagane	Hash list subskrypcyjnej
<b>limit</b>	Opcjonalne	Ilość pobranych akcji subskrybenta (domyślnie 10)

Przykładowy request:

```
{  
  "email":"tester1@freshmail.pl"  
  "list":"4zcnmd2ski"  
  "limit":"30"  
}
```

Zwracane dane:

<b>name</b>	Nazwa kampanii
<b>email_topic</b>	Temat kampanii
<b>scheduled_sent</b>	Data i czas wysłania kampanii (format: rrrr-mm-dd hh:ii:ss)
<b>state</b>	Status subskrybenta (dostarczony, odbicie miękkie, odbicie twarde)
<b>opens_count</b>	Ilość otwarć subskrybenta
<b>clicks_count</b>	Ilość kliknięć subskrybenta

Kody błędów:

<b>1311</b>	Adres email jest niepoprawny
<b>1312</b>	Lista subskrypcyjna nie istnieje
<b>1313</b>	Subskrybent nie istnieje w podanej liście subskrypcyjnej
<b>1314</b>	Wartość atrybutu limit jest niepoprawna
<b>1315</b>	Subskrybent nie posiada akcji

Przykład odpowiedzi niepoprawnej:

```
{ "status": "ERROR",  
  "errors": [{  
    "message": "Subscriber doesn't exist",  
    "code": 1313  
  }]  
}
```

Przykład odpowiedzi poprawnej:

```
{ "status": "OK",  
  "data": [{  
    "name": "4 kampania wtorkowa z dnia 30.07.2013",  
    "email_topic": "Jesienna promocja",  
    "scheduled_sent": "2013-07-30 21:37:42",  
    "state": "DELIVERED",  
    "opens_count": "1",  
    "clicks_count": "2"  
  }]  
}
```

## Subskrybenci – zarządzanie wieloma subskrybentami w pojedynczym żądaniu

Za pomocą poniższych metod można wykonywać dokładnie te same działania co przy zarządzaniu pojedynczymi subskrybentami, ale w żądaniu można przestać do 100 subskrybentów naraz.

### Dodawanie wielu subskrybentów

Metoda dodaje wielu subskrybentów jeśli nie było ich w liście subskrypcyjnej lub aktualizuje ich status i dane jeśli nie był w statusie „**Aktywny**” oraz „**Do aktywacji**”. W przypadku dodania tylko części z adresów (np. jeden z adresów jest niepoprawny) zostaje zwrócony kod 200 oraz dane dotyczące błędnego/błędnych adresów.

Wywołanie: **POST /rest/subscriber/addMultiple**

Dane w POST:

<b>subscribers</b>	Wymagane	Tablica z adresami email i polami dodatkowymi w postaci { "email": "adres_email", "custom_fields": { "tag personalizacji pola 1": "wartość", "tag personalizacji pola 1": "wartość" } } Pole "custom_fields" nie jest obowiązkowe.
<b>list</b>	Wymagane	Hash list subskrypcyjnej
<b>state</b>	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste zostanie nadany status „ <b>Do aktywacji</b> ”, lista dostępnych statusów w dziale „ <b>Kody Statusów Subskrybentów</b> ”
<b>confirm</b>	Opcjonalne	Czy do subskrybenta ma być wysłany email potwierdzający. Możliwe wartości to <b>0</b> oraz <b>1</b> . Jeśli pole będzie puste zostanie wysłany email z potwierdzeniem. To pole jest ignorowane (email nie zostanie wysłany) jeśli nadamy subskrybentowi status inny niż „ <b>Do aktywacji</b> ”.

Przykładowy request:

```
{"list": "id_hash",  
  "subscribers": [ {"email": "tester1@freshmail.pl",  
                   "custom_fields": {"imie": "Hanna",  
                                     "kod_promocyjny": "abcde"} },  
                  {"email": "tester2@freshmail.pl",  
                   "custom_fields": {"imie": "Anna",
```



```
        "kod_promocyjny":"efgh"}},
{"email":"tester3@freshmail.pl",
 "custom_fields":{"imie":"Agata",
 "kod_promocyjny":"wxyz"} } ] }
```

#### Kody błędów:

- |             |  |
|-------------|--|
| <b>1331</b> | Nie udało się dodać/aktualizować żadnego subskrybenta  |
| <b>1332</b> | Lista subskrypcyjna nie istnieje lub brak hash'a listy |
| <b>1335</b> | Próbowano nadać niepoprawny status subskrybenta        |
| <b>1336</b> | Nie przekazano adresów email do dodania                |
| <b>1399</b> | Za dużo subskrybentów w jednym żądaniu – limit 100     |

#### Przykład częściowych błędów, wysłany request:

```
{"list":"4zcnmd2ski",
 "subscribers":[{"email":"poprawny@adres.email.pl"},
 {"email":"niepoprawny adres email"}]}
```

#### Zwracane dane (kod odpowiedzi 200):

```
{"status":"OK",
 "data":{"
 "inserted":1,
 "not_inserted":1,
 "errors":[{"email":"niepoprawny adres email",
 "error":"Email address not correct",
 "code":1301}]}}
```

Kody błędów zwracane dla poszczególnych subskrybentów są identyczne jak kody błędów dla metody **/rest/subscriber/add**

## Modyfikowanie danych wielu subskrybentów

Metoda aktualizuje status i dane wielu subskrybentów

Wywołanie: **POST /rest/subscriber/editMultiple**

Dane w POST:

<b>subscribers</b>	Wymagane	Tablica z adresami email i polami dodatkowymi w postaci <pre>{ "email": "adres_email",   "custom_fields":     { "tag personalizacji pola 1": "wartość",       "tag personalizacji pola 1": "wartość" } }</pre> Pole "custom_fields" nie jest obowiązkowe.
<b>list</b>	Wymagane	Hash list subskrypcyjnej
<b>state</b>	Opcjonalne	Status jaki chcemy nadać subskrybentowi, jeśli pole będzie puste aktualny status subskrybenta nie zostanie zmieniony

Kody błędów:

<b>1331</b>	Nie udało się dodać/aktualizować żadnego subskrybenta
<b>1332</b>	Lista subskrypcyjna nie istnieje lub brak hash'a listy
<b>1335</b>	Próbowano nadać niepoprawny status subskrybenta
<b>1336</b>	Nie przekazano adresów email do zaktualizowania
<b>1399</b>	Za dużo subskrybentów w jednym żądaniu – limit 100

Zwracane dane/informacje o błędach są identyczne jak w przypadku metody  
**/rest/subscriber/addMultiple**

## Modyfikowanie pól dodatkowych wszystkich subskrybentów

Metoda aktualizuje wybrane pole dodatkowe i ustawia ustaloną wartość dla wszystkich subskrybentów

Wywołanie: **POST /rest/subscriber/updateFieldValue**

Dane w POST:

<b>listHash</b>	Wymagane	Hash list subskrypcyjnej
<b>tag</b>	Wymagane	Tag pola dodatkowego (np. kod_promocyjny)
<b>value</b>	Wymagane	Wartość jak zostanie ustawiona w danym polu dodatkowym dla wszystkich subskrybentów
<b>url</b>	Wymagane	Adres www na który ma zostać wysłana odpowiedź po zakończeniu procesu aktualizacji

Zwracane dane:

<b>id</b>	Id procesu
-----------	------------

Kody błędów:

<b>1381</b>	Pole dodatkowe nie istnieje
<b>1382</b>	Brak hasha listy lub tag'a pola dodatkowego
<b>1383</b>	Niepoprawny adres url
<b>1384</b>	Zbyt duża liczba znaków w polu „value” - max. 255
<b>1385</b>	Pole „value” jest wymagane
<b>1386</b>	Proces aktualizacji tego pola wciąż trwa, poczekaj na jego zakończenie

Przykład odpowiedzi niepoprawnej:

```
{
  "status": "ERROR",
  "errors": [
    {
      "message": "Niepoprawny adres url",
      "code": 1383
    }
  ]
}
```

Przykład odpowiedzi poprawnej (kod odpowiedzi 200):

```
{
  "status": "OK",
  "data": {
    "id": 5
  }
}
```

```
}}
```

Po procesie aktualizacji zostanie wysłane żądanie na wskazany adres URL. Odpowiedz będzie żądaniem typu POST, w treści żądania znajdą się następujące dane:

<b>id</b>	Id procesu
<b>updated</b>	Ilość zaktualizowanych pól subskrybentów

Przykładowe żądanie:

```
{ "status": "OK",  
  "data": {  
    "id": 5,  
    "updated": 150  
  }  
}
```

W odpowiedzi skrypt oczekuje kody se statusem „OK”:

```
{ "status": "OK" }
```

Jeśli powyższa odpowiedz nie będzie poprawna, skrypt ponowi kilka razy próbę komunikacji, przy większej ilości błędów nastąpi zaniechanie wysyłania kolejnych żądań.

## Pobieranie danych wielu subskrybentów

W przypadku gdy request jest częściowo niepoprawny (np. część adresów email nie istnieje) lub całkowicie niepoprawny (np. żaden z adresów email nie istnieje) zostaje zwrócony kod 200 oraz dane dotyczące błędnego/błędnych adresów.

Wywołanie: POST **/rest/subscriber/getMultiple**

Dane w GET:

<b>subscribers</b>	Wymagane	Tablica z adresami email użytkowników: [{"email": " <a href="mailto:tester1@freshmail.pl">tester1@freshmail.pl</a> "}, {"email": " <a href="mailto:tester2@freshmail.pl">tester2@freshmail.pl</a> "}, {"email": " <a href="mailto:tester3@freshmail.pl">tester3@freshmail.pl</a> "}]
<b>list</b>	Wymagane	Hash list subskrypcyjnej

Zwracane dane:

<b>email</b>	Adres email subskrybenta
<b>state</b>	Status subskrybenta (zgodny z opisem w dziale „ <b>Kody Statusów Subskrybentów</b> ”)
<b>custom_fields</b>	Pola dodatkowe wraz z wartościami
<b>errors</b>	Tablica z błędami (w przypadku gdy któryś z adresów nie istnieje)

Kody błędów:

<b>1341</b>	Lista subskrypcyjna nie istnieje
<b>1342</b>	Tablica z subskrybentami jest pusta
<b>1399</b>	Za dużo subskrybentów w jednym żądaniu – limit 100
<b>1336</b>	Nie przekazano adresów odbiorców do pobrania

## Kasowanie wielu subskrybentów

Metoda kasuje subskrybentów z listy subskrypcyjnej

Wywołanie: **POST /rest/subscriber/deleteMultiple**

Dane w POST:

<b>subscribers</b>	Wymagane	Tablica z adresami email użytkowników: [ {"email": " <a href="mailto:tester1@freshmail.pl">tester1@freshmail.pl</a> "}, {"email": " <a href="mailto:tester2@freshmail.pl">tester2@freshmail.pl</a> "}, {"email": " <a href="mailto:tester3@freshmail.pl">tester3@freshmail.pl</a> "} ]
<b>list</b>	Wymagane	Hash list subskrypcyjnej

Kody błędów:

- 1351** Lista subskrypcyjna nie istnieje
- 1352** Żaden z subskrybentów nie istnieje/nie został skasowany
- 1399** Za dużo subskrybentów w jednym żądaniu – limit 100

Zwracane dane/informacje o błędach są identyczne jak w przypadku metody  
`/rest/subscriber/addMultiple`

## Dodawanie subskrybentów do listy odbiorców blokowanych

Metoda dodaje adresy subskrybentów do listy odbiorców blokowanych.

Wywołanie: POST **/rest/subscriber/addBlocks**

Dane w POST:

<b>emails</b>	Wymagane	Tablica z adresami email użytkowników: { "emails": [ " <a href="mailto:tester1@freshmail.pl">tester1@freshmail.pl</a> ", " <a href="mailto:tester2@freshmail.pl">tester2@freshmail.pl</a> ", " <a href="mailto:tester3@freshmail.pl">tester3@freshmail.pl</a> " ] } ]}
---------------	----------	--

Zwracane dane:

<b>stats['added']</b>	Ilość dodanych odbiorców
<b>stats['existing']</b>	Ilość już istniejących odbiorców
<b>stats['incorrect']</b>	Ilość błędnych odbiorców
<b>Incorrect_mails</b>	Tablica z błędnymi adresami (w przypadku gdy jakiś mail był błędny)

Kody błędów:

<b>1301</b>	Adresy odbiorców są niepoprawne
<b>1336</b>	Nie przekazano adresów odbiorców do dodania
<b>1343</b>	Format przekazanych danych jest niepoprawny

## Kasowanie subskrybentów z listy odbiorców blokowanych

Metoda kasuje subskrybentów z listy odbiorców blokowanych. Akcja możliwa jedynie przy adresach zablokowanych własnoręcznie.

Wywołanie: POST **/rest/subscriber/removeBlocks**

Dane w POST:

<b>emails</b>	Wymagane	Tablica z adresami email użytkowników: { "emails": [ " <a href="mailto:tester1@freshmail.pl">tester1@freshmail.pl</a> ", " <a href="mailto:tester2@freshmail.pl">tester2@freshmail.pl</a> ", " <a href="mailto:tester3@freshmail.pl">tester3@freshmail.pl</a> " ] } ]}
---------------	----------	--

Zwracane dane:

<b>stats['removed']</b>	Ilość skasowanych odbiorców
<b>stats['not_existing']</b>	Ilość nieistniejących odbiorców
<b>stats['incorrect']</b>	Ilość błędnych odbiorców

Kody błędów:

<b>1336</b>	Nie przekazano adresów odbiorców do dodania
<b>1343</b>	Format przekazanych danych jest niepoprawny



## Tworzenie kont

Metoda tworzy konto użytkownika. Można utworzyć zarówno konto główne jak i konto podrzędne.

Wywołanie: **POST /rest/account/create**

Dane w POST:

<b>login</b>	Wymagane	Login nowego użytkownika (adres email)
<b>password</b>	Wymagane	Hasło którym użytkownik będzie się logował
<b>firstname</b>	Wymagane	Imię użytkownika
<b>lastname</b>	Wymagane	Nazwisko użytkownika
<b>company</b>	Opcjonalne	Nazwa firmy
<b>phone</b>	Wymagane	Numer telefonu
<b>activation_mail</b>	Opcjonalne	Czy ma być wysłany mail aktywacyjny (domyślnie true)
<b>activation</b>	Opcjonalne	Czy konto ma wymagać aktywacji (domyślnie true)
<b>child_account</b>	Opcjonalne	Czy konto ma być kontem podrzędnym (domyślnie true)

Zwracane dane:

<b>hash</b>	Hash nowo założonego konta
<b>api_key</b>	Api key nowo założonego konta
<b>api_secret</b>	Secret nowo założonego konta

Kody błędów:

<b>1501</b>	Brak loginu (adres email)
<b>1502</b>	Nieprawidłowy login
<b>1503</b>	Login już istnieje w systemie
<b>1504</b>	Brak hasła
<b>1505</b>	Hasło jest za słabe - musi zawierać co najmniej 8 znaków - w tym jedną literę, jedną liczbę oraz jeden znak specjalny
<b>1506</b>	Brak imienia

- 1507** Brak nazwiska
- 1509** Brak telefonu
- 1510** Wystąpił błąd podczas tworzenia konta
- 1511** Konto nie zostało utworzone.
- 1512** Nie udało się przypisać konta do aplikacji
- 1513** Brak uprawnień do tworzenia kont.

## Listy subskrypcyjne - zarządzanie

Kontroler służy do zarządzania listami subskrypcyjnymi. Umożliwia ich dodawanie, modyfikowanie, listowanie oraz usuwanie.

### Tworzenie list

Funkcja służy do tworzenia list subskrypcyjnych. Zwraca hash listy, którym można się dalej postugiwać.

Wywołanie: **/rest/subscribers\_list/create**

Dane:

<b>name</b>	Wymagane	Nazwa listy subskrypcyjnej
<b>description</b>	Opcjonalne	Opis listy subskrypcyjnej
<b>custom_fields</b>	Opcjonalne	Tablica definiująca pola dodatkowe (wedle opisu z punktu <b>Tworzenie pól dodatkowych w listach</b> )

Zwracane dane:

<b>hash</b>	Hash nowej listy subskrypcyjnej
<b>custom_fields</b>	Tablica z pełnymi opisami utworzonych pól – jeśli miały być utworzone.

Kody błędów:

- 1601** Pusta wartość w polu name
- 1602** Nie udało się utworzyć listy subskrypcyjnej
- 1603** Błędna lista pól dodatkowych.
- 1604** Tag personalizacji powinien składać się jedynie z liter, cyfr i znaku podkreślenia.
- 1605** Nieprawidłowy typ pola.
- 1606** Osiągnięto maksymalną ilość list na koncie

## Modyfikowanie list

Funkcja służy do modyfikowania list subskrypcyjnych.

Wywołanie: **/rest/subscribers\_list/update**

Dane:

<b>hash</b>	Wymagane	Hash listy subskrypcyjnej
<b>name</b>	Wymagane	Nazwa listy subskrypcyjnej
<b>description</b>	Opcjonalne	Opis listy subskrypcyjnej

Kody błędów:

- 1611** Pusta wartość w polu name lub brak parametru name
- 1604** Brak uprawnień do listy

## Usuwanie list

Funkcja służy do usuwania list subskrypcyjnych.

Wywołanie: **/rest/subscribers\_list/delete**

Dane:

<b>hash</b>	Wymagane	Hash listy subskrypcyjnej
-------------	----------	---------------------------

Kody błędów:

**1604** Brak uprawnień do listy

**1605** Nie udało się usunąć listy subskrypcyjnej

## Pobieranie danych o listach

Funkcja służy do listowania wszystkich list subskrypcyjnych. Zwraca tablice z danymi wszystkich list do których konto ma uprawnienia.

Wywołanie: **`/rest/subscribers_list/lists`**

Zwracane dane:

<b>hash</b>	Hash listy subskrypcyjnej
<b>name</b>	Nazwa listy subskrypcyjnej
<b>description</b>	Opis listy subskrypcyjnej
<b>creation_date</b>	Data utworzenia listy subskrypcyjnej
<b>subscribers_number</b>	Liczba aktywnych subskrybentów
<b>list_type</b>	Typ listy, możliwe wartości to: „single opt-in” oraz „double opt-in”

## Tworzenie pól dodatkowych w listach

Funkcja służy do tworzenia pól dodatkowych w listach subskrypcyjnych

Wywołanie: **/rest/subscribers\_list/addField**

Dane:

<b>hash</b>	Wymagane	Hash listy subskrypcyjnej
<b>name</b>	Wymagane	Nazwa pola w liście
<b>tag</b>	Opcjonalne	Tag pod którym wartości z tego pola będą dostępne w mailach (w przypadku braku jest tworzony na podstawie nazwy pola)
<b>type</b>	Opcjonalne	Typ pola – 0 tekstowe, 1 numeryczne (domyślnie tekstowe)

Zwracane dane:

<b>id_hash</b>	Hash pola dodatkowego
<b>field_name</b>	Nazwa pola dodatkowego
<b>personalization_tag</b>	Tag pod którym wartości z tego pola będą dostępne w mailach
<b>field_type</b>	Typ pola 0 – tekstowe, 1 - numeryczne

Kody błędów:

<b>1622</b>	Błędny lub pusty hash listy subskrypcyjnej
<b>1623</b>	Błędna lub pusta nazwa pola
<b>1624</b>	Błędna nazwa tagu
<b>1625</b>	Błędny typ pola
<b>1626</b>	Pole dodatkowe o takim tagu personalizacji już istnieje

## Wyświetlenie listy pól dodatkowych w podanej liście

Funkcja służy do wyświetlenia wszystkich pól dodatkowych w wybranej liście subskrypcyjnej

Wywołanie: **/rest/subscribers\_list/getFields**

Dane:

<b>hash</b>	Wymagane	Hash listy subskrypcyjnej
-------------	----------	---------------------------

Zwracane dane:

<b>hash</b>	Hash pola dodatkowego
<b>name</b>	Nazwa pola dodatkowego
<b>tag</b>	Tag pod którym wartości z tego pola będą dostępne w mailach
<b>type</b>	Typ pola

Kody błędów:

<b>1632</b>	Błędny lub pusty hash listy subskrypcyjnej
-------------	--



## Spamtesty

Kontroler służy do sprawdzania maili pod kątem prawdopodobieństwa wpadnięcia wiadomości do spamu.

### Sprawdzenie zgodności z filtrami SpamAssassin

Medota służy do przetestowania ile punktów spamowych oraz za co dostanie Twoja wiadomość w filtrze SpamAssassin.

**Pamiętaj: Nie wszystkie systemy pocztowe korzystają z tego oprogramowania, wynik jest poglądowy i przy innej konfiguracji programów pocztowych może się różnić od pokazanego w tym teście. Wynik pozytywny (poniżej 5.0 punktów spamowych NIE oznacza, że inne filtry antyspamowe nie zatrzymają twojej wiadomości).**

Wywołanie: `/rest/spam_test/check`

Dane:

<b>subject</b>	Wymagane	Temat maila
<b>from</b>	Opcjonalne	Adres nadawcy
<b>from_name</b>	Opcjonalne	Nazwa nadawcy
<b>html</b>	Opcjonalne *)	Treść html
<b>text</b>	Opcjonalne *)	Treść tekstowa

\*) Co najmniej jedno z wyróżnionych pól musi być wypełnione

Zwracane dane:

<b>tests</b>	Lista testów antyspamowych których mail nie przeszedł
<b>score</b>	Sumaryczna punktacja za testy antyspamowe

Kody błędów:

<b>1901</b>	Brak tematu
<b>1902</b>	Niepoprawny adres nadawcy „from”
<b>1904</b>	Brak treści maila
<b>1906</b>	Błąd systemu sprawdzania.
<b>1907</b>	Przekroczono limit testów w ciągu jednego dnia